LEARNING WITH REFLECTIVE LIKELIHOODS

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine learning systems have achieved state-of-the-art results in many domains. They are usually trained using the maximum likelihood principle. However maximum likelihood learning can lead to poor learned representations of high dimensional data. For example this is manifested in deep generative latent variable models where the latent variables and their associated observations are driven independent from each other. We identify a peculiarity in maximum likelihood learning that causes this problem of poor learned representations. We then propose a new learning criterion for better representation learning. The proposed criterion relies on simultaneously maximizing the likelihood of the data and minimizing what we term the *reflective likelihood* of the data. We study this new criterion both theoretically and empirically and show improved performance on image classification under imbalance and text modeling with deep generative latent variable models.

1 INTRODUCTION

We are concerned with learning in probabilistic models where we make assumptions about dependencies between variables and use data to learn the dependencies. These dependencies can be expressed using a probabilistic graphical model[1] (Koller et al., 2009). These models often have some practically desirable properties for learning; for example the conditional conjugacy in the model of Blei et al. (2003). However this practical advantage comes at the cost of less expressivity. Recently much focus has been devoted to devising models parameterized by deep neural networks (Neal, 1992; Dayan et al., 1995; MacKay & Gibbs, 1999; LeCun et al., 2015). These are very expressive models that have achieved state-of-the-art performance on many domains (LeCun et al., 1995; Hochreiter & Schmidhuber, 1997; Sutskever et al., 2014).

Learning is not merely about specifying a model; it also involves specifying a criterion—an objective function that informs us of how well we are fitting the data with our model. There are several desiderata for such a criterion. In addition to good generalization abilities, we want the optimization of a criterion to be stable, statistically efficient, and convergent. Many learning objectives have been proposed (Fisher, 1997; Tishby et al., 2000; Gutmann & Hyvärinen, 2010; Goodfellow et al., 2014). In this paper we focus on maximum likelihood.

Learning models parameterized by deep neural networks using maximum likelihood has led to many successes in density estimation, variational inference, and text and image generation (Dinh et al., 2016; Kingma & Dhariwal, 2018; Kingma & Welling, 2013; Rezende et al., 2014; Oord et al., 2016). However maximum likelihood learning of deep models often causes the problem that we call *input forgetting*—so called because it corresponds to ignoring the input. We refer to an input here as any variable being conditioned upon—for example a covariate in supervised learning or a latent variable in deep generative latent variable models. This problem is referred to as *latent variable collapse* in the context of latent variable models and has been discussed in several works (Bowman et al., 2015; Zhao et al., 2017; Dieng et al., 2018a). Input forgetting makes posterior inference in deep generative models very difficult. It also occurs when learning with Restricted Boltzmann Machines (RBMs) where all the hidden units of the RBM easily learn to capture the bias in the visible units thus becoming useless (Cho et al., 2011).

Contributions. We identify a peculiarity in maximum likelihood learning that causes the input forgetting problem in Section 2.1. We then propose a new learning criterion to mitigate this issue. The proposed criterion simultaneously maximizes the likelihood of the data while minimizing what

[1] Also called a Bayesian network.

we call the *reflective likelihood* of the data. Maximizing the likelihood helps find parameters that fit the data. Minimizing the reflective likelihood favors those parameters—among all the parameters that can explain the data—for which outputs are likely only when accounting for the input. We define and outline the proposed objective for both supervised and unsupervised learning with latent variable models in Section 2.2. In this same Section 2.2 we make some connections to ranking losses when using a particular form of the weight used to trade-off these two forms of likelihoods. Finally in Section 4 we show improved performance on image classification under imbalance and latent variable text modeling when using the proposed learning criterion.

2 LEARNING WITH REFLECTIVE LIKELIHOODS

In this section we first identify and explain a peculiarity of maximum likelihood learning. We attribute failures of maximum likelihood learning when it comes to representation learning to this peculiarity. We then propose a new learning criterion—for both supervised and unsupervised learning—to mitigate this problem. Finally we provide some connections to ranking losses.

2.1 A PECULIARITY OF MAXIMUM LIKELIHOOD LEARNING

We consider a supervised learning setting where there are inputs $\mathbf{x}$ and their associated outputs $\mathbf{y}$. These (input, output) pairs are drawn from an unknown distribution $p_{\text{data}}(\mathbf{x}, \mathbf{y})$ that can be factorized using the chain rule as

$$p_{\text{data}}(\mathbf{x}, \mathbf{y}) = p_{\text{data}}(\mathbf{x}) p_{\text{data}}(\mathbf{y} \mid \mathbf{x}).$$

In supervised learning we are interested in estimating the unknown conditional distribution $p_{\text{data}}(\mathbf{y} \mid \mathbf{x})$. There are many approaches to this problem. We consider the traditional approach of positing a family of distributions $\mathcal{P} = \{p_\theta : \theta \in \Theta\}$ indexed by a set of parameters Θ and finding the optimal parameter $\theta^* \in \Theta$ that best explains the observed data. The maximum likelihood principle achieves this by solving the optimization problem

$$\theta^* = \arg\max_{\theta \in \Theta} \; \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} \mathbb{E}_{\mathbf{y} \sim p_{\text{data}}(\mathbf{y} \mid \mathbf{x})} \left[\log p_\theta(\mathbf{y} \mid \mathbf{x}) \right], \tag{1}$$

where the expectations are estimated using the observed data. The optimization in Eq. 1 is simple and leads to parameter estimators with several desirable theoretical properties such as efficiency and asymptotic normality. This maximum likelihood learning procedure has been the workhorse behind many successes of machine learning.

However using data to optimize Eq. 1 can be achieved in two different ways. The first way is to minimize the Kullback-Leibler (KL) divergence between $p_{\text{data}}(\mathbf{y} \mid \mathbf{x})$ and $p_\theta(\mathbf{y} \mid \mathbf{x})$,

$$\theta^* = \arg\max_{\theta \in \Theta} \; \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} \left[\text{KL} \left(p_{\text{data}}(\mathbf{y} \mid \mathbf{x}) \,\|\, p_\theta(\mathbf{y} \mid \mathbf{x}) \right) \right]. \tag{2}$$

The second way to optimize Eq. 1 using data, however, is to match the marginal distributions over the output $\mathbf{y}$, i.e., minimize the KL divergence between $p_{\text{data}}(\mathbf{y})$ and $p_\theta^{\text{refl}}(\mathbf{y})$:

$$\theta^* = \arg\max_{\theta \in \Theta} \; \text{KL} \left(p_{\text{data}}(\mathbf{y}) \,\|\, p_\theta^{\text{refl}}(\mathbf{y}) \right) \tag{3}$$

$$\text{where} \quad p_{\text{data}}(\mathbf{y}) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} p_{\text{data}}(\mathbf{y} \mid \mathbf{x}) \quad \text{and} \quad p_\theta^{\text{refl}}(\mathbf{y}) = \mathbb{E}_{\mathbf{x}' \sim p_{\text{data}}(\mathbf{x}')} \left[p_\theta(\mathbf{y} \mid \mathbf{x}') \right].$$

We refer to $p_\theta^{\text{refl}}(\mathbf{y})$ as the *reflective likelihood*—so called because it can be interpreted as projecting $\mathbf{y}$ onto all possible inputs $\mathbf{x}'$ and measuring how likely it is. This is in contrast to the *dependent likelihood* $p_\theta(\mathbf{y} \mid \mathbf{x})$ which considers a single correctly paired input $\mathbf{x}$.

Maximizing Eq. 1 can be achieved by maximizing either Eq. 2 or Eq. 3 or both. The problem with maximum likelihood learning is that we do not have control over the trade-off between these two optimization procedures.

Optimizing Eq. 1 by following the procedure of Eq. 3 fits a marginal model of $\mathbf{y}$ to the data. As a result the corresponding parameters θ do not fully capture the dependencies between the inputs $\mathbf{x}$ and outputs $\mathbf{y}$ present in the data. The inputs are not predictive of the outputs.

This independence problem between variables can also happen in the unsupervised learning setting. The analysis above can be replicated in the unsupervised learning case by considering latent variables $\mathbf{z}$ and data $\mathbf{y}$.

Ultimately we want our learning procedure to follow the *dependence path*—the subspace in Θ for which inputs and outputs are dependent. However this dependence path is unknown to us; there is nothing in Eq. 1 that guides learning to follow this dependence path instead of following Eq. 3—the *independence path*. In fact when the model has enough capacity to capture the marginal this independence behavior takes place. This problem is prominent with deep neural network-based models and manifests itself in various ways depending on the application. For example in neural machine translation this problem leads to translations that don't account for the input. In conversation models it causes the problem of lack of diversity in the generated responses. In deep generative latent variable models it causes the problem known as *latent variable collapse* where the latent variables do not encode any information about the data.

In the next section we propose a new learning criterion that favors the dependence path.

2.2 Guiding maximum likelihood learning with reflective likelihoods

We now propose a learning criterion for both supervised and unsupervised learning that penalizes the independence behavior induced by Eq. 3.

Supervised learning. We propose to regularize maximum likelihood learning by minimizing the reflective likelihood. That is we propose to maximize,

$$\mathcal{L}_{\text{RLL}} = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\mathbb{E}_{\mathbf{y}\sim p_{\text{data}}(\mathbf{y}\,|\,\mathbf{x})}\left[\log p_\theta(\mathbf{y}\,|\,\mathbf{x}) - \alpha(\mathbf{x},\mathbf{y},\theta)\log p_\theta^{\text{refl}}(\mathbf{y})\right], \tag{4}$$

where $\alpha(\mathbf{x},\mathbf{y},\theta) > 0$—which we will describe shortly—is a coefficient that controls the penalty imposed by the reflective likelihood.

The proposed criterion in Eq. 4—just like the maximum likelihood objective in Eq. 1—can be approximated by replacing the unknown data distribution with the empirical data distribution. This leads to the following objective function:

$$\hat{\mathcal{L}}_{\text{RLL}} = \frac{1}{N}\sum_{(\mathbf{x}_n,\mathbf{y}_n)\in\mathcal{D}}\left[\log p_\theta(\mathbf{y}_n\,|\,\mathbf{x}_n) - \alpha(\mathbf{x}_n,\mathbf{y}_n,\theta)\log \hat{p}_\theta^{\text{refl}}(\mathbf{y}_n)\right],$$

where $\mathcal{D}$ denotes the observed data and N is the total number of observations in $\mathcal{D}$. Here $\hat{p}_\theta^{\text{refl}}$ is an empirical estimate of the true reflective likelihood[2],

$$\hat{p}_\theta^{\text{refl}}(\mathbf{y}) \approx \frac{1}{M}\sum_{m=1}^{M} p_\theta(\mathbf{y}\,|\,\mathbf{x}^m) \quad \text{where} \quad \mathbf{x}^1,\ldots,\mathbf{x}^M \sim \mathcal{D}. \tag{5}$$

The major motivation behind our proposed method is to encourage parameter settings for which the following holds for any pair $(\mathbf{x},\mathbf{y})$:

$$\log p_\theta(\mathbf{y}\,|\,\mathbf{x}) > \log p_\theta^{\text{refl}}(\mathbf{y}) \text{ a.e.}$$

In other words, when learning proceeds, we want to encourage settings of the parameters θ for which $\mathbf{y}$ is more likely when conditioning on $\mathbf{x}$ than when averaging all the conditional distributions of $\mathbf{y}$ given all possible inputs.

Unsupervised learning with latent variable models[3]. We now extend the objective proposed above to unsupervised learning with latent variable models. We position ourselves in the setting where there are global parameters θ and one latent variable $\mathbf{z}$ for every observation $\mathbf{y}$. An observation $\mathbf{y}$ is generated by first drawing a latent variable $\mathbf{z}$ from some prior distribution $p(\mathbf{z})$—that we assume fixed—and then sampling $\mathbf{y}$ from the conditional distribution of $\mathbf{y}$ given $\mathbf{z}$. This conditional distribution $p_\theta(\mathbf{y}\,|\,\mathbf{z})$ is parameterized by θ. We are concerned with learning the parameters θ in the presence of the latent variables. Maximum likelihood corresponds to maximizing

$$\mathcal{L}_{\text{MLE}} = \mathbb{E}_{\mathbf{y}\sim p_{\text{data}}(\mathbf{y})}\log p_\theta(\mathbf{y}) = \mathbb{E}_{\mathbf{y}\sim p_{\text{data}}(\mathbf{y})}\left[\log\int_z p_\theta(\mathbf{y}\,|\,\mathbf{z})p(\mathbf{z})\mathrm{d}\mathbf{z}\right]. \tag{6}$$

[2]The true reflective likelihood p_θ^{refl} is intractable for large datasets because for each observation it involves all the other elements in the dataset.

[3]The method we devise here is also applicable when there are extra features $\mathbf{x}$ to condition on. Simply add this conditioning on all the densities.

The integral above is often intractable. Existing solutions include Markov chain Monte Carlo methods and approximation methods such as importance sampling and variational inference. However the problem we identified in Section 2.1 is also present in this setting. In fact when $p_\theta(\mathbf{y} \mid \mathbf{z})$ is represented as a powerful deep neural network the input $\mathbf{z}$ is often ignored. This is because the objective in Eq. 6 can be maximized by maximizing

$$\mathbb{E}_{\mathbf{y}\sim p_{\text{data}}(\mathbf{y})} \log p_\theta^{\text{refl}}(\mathbf{y}) = \mathbb{E}_{\mathbf{y}\sim p_{\text{data}}(\mathbf{y})} \left[\log \mathbb{E}_{\mathbf{y}'\sim p_{\text{data}}(\mathbf{y}')} \left(\int_{\mathbf{z}} p_\theta(\mathbf{y} \mid \mathbf{z}) p_\theta(\mathbf{z} \mid \mathbf{y}') \mathrm{d}\mathbf{z} \right) \right]. \tag{7}$$

The reflective likelihood $p_\theta^{\text{refl}}(\mathbf{y})$ is a marginal distribution over $\mathbf{y}$ that corresponds to a projection of $\mathbf{y}$ on all possible latent variables $\mathbf{z}$. These latent variables emerge from drawing an observation $\mathbf{y}'$ from the data and sampling $\mathbf{z}$ from the true posterior $p_\theta(\mathbf{z} \mid \mathbf{y}')$.

To promote the dependence path—the one where a latent $\mathbf{z}$ and its associated observation $\mathbf{y}$ are strongly dependent on each other—we propose to maximize

$$\mathcal{L}_{\text{RLL}} = \mathbb{E}_{\mathbf{y}\sim p_{\text{data}}(\mathbf{y})} \Big[\log p_\theta(\mathbf{y}) - \alpha(\mathbf{y}, \theta) \log p_\theta^{\text{refl}}(\mathbf{y}) \Big].$$

Replacing $p_\theta(\mathbf{y})$ and $p_\theta^{\text{refl}}(\mathbf{y})$ using their expressions in Eq. 6 and Eq. 7 we have

$$\mathcal{L}_{\text{RLL}} = \mathbb{E}_{\mathbf{y}\sim p_{\text{data}}(\mathbf{y})} \Big[\log \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})} \left[p_\theta(\mathbf{y} \mid \mathbf{z}) \right] - \alpha(\mathbf{y}, \theta) \log \mathbb{E}_{\mathbf{y}'\sim p_{\text{data}}(\mathbf{y}')} \mathbb{E}_{\mathbf{z}\sim p_\theta(\mathbf{z} \mid \mathbf{y}')} \left[p_\theta(\mathbf{y} \mid \mathbf{z}) \right] \Big] \tag{8}$$

The function $\alpha(\mathbf{y}, \theta)$ has the same role as in the supervised learning case; it controls the level of penalization of the independence path. We will discuss it shortly.

The expectations in Eq. 8 are intractable. In this paper, we propose to approximate them using importance weighting. For that we define a parametric proposal distribution $q_\phi(\mathbf{z} \mid \mathbf{y})$—we make the conditioning on $\mathbf{y}$ explicit to account for recognition networks as proposal distributions. We now write the expectations as

$$\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})} \left[p_\theta(\mathbf{y} \mid \mathbf{z}) \right] \approx \hat{p}_\theta(\mathbf{y}) = \frac{1}{K} \sum_{k=1}^{K} \omega(\mathbf{y}, \mathbf{z}^k) p_\theta(\mathbf{y} | \mathbf{z}^k) \quad \text{where} \quad \mathbf{z}^k \sim q_\phi(\mathbf{z} \mid \mathbf{y});$$

$$\text{and} \quad \mathbb{E}_{\mathbf{y}'\sim p_{\text{data}}(\mathbf{y}')} \mathbb{E}_{\mathbf{z}\sim p_\theta(\mathbf{z} \mid \mathbf{y}')} \left[p_\theta(\mathbf{y} \mid \mathbf{z}) \right] \approx \mathbb{E}_{\mathbf{y}'\sim p_{\text{data}}(\mathbf{y}')} \left(\sum_{k=1}^{K} v(\mathbf{y}', \mathbf{z}^k) p_\theta(\mathbf{y} \mid \mathbf{z}^k) \right)$$

where $\mathbf{z}^k \sim q_\phi(\mathbf{z} \mid \mathbf{y}')$. The expensive expectation over $p_{\text{data}}(\mathbf{y}')$ is dealt with as before by using a small random subset of training examples—in Section 4 we use 5 samples. The importance weights $\omega(\mathbf{y}, \mathbf{z}^k)$ and $v(\mathbf{y}', \mathbf{z}^k)$ are computed as

$$\omega(\mathbf{y}, \mathbf{z}^k) = \exp(\tilde{\omega}(\mathbf{y}, \mathbf{z}^k)) \quad \text{and} \quad \tilde{\omega}(\mathbf{y}, \mathbf{z}^k) = \log p(\mathbf{z}^k) - \log q_\phi(\mathbf{z}^k \mid \mathbf{y}).$$

$$v(\mathbf{y}', \mathbf{z}^k) = \frac{\exp(\tilde{v}(\mathbf{y}', \mathbf{z}^k))}{\sum_{s=1}^{K} \exp(\tilde{v}(\mathbf{y}', \mathbf{z}^s))} \quad \text{and} \quad \tilde{v}(\mathbf{y}', \mathbf{z}^k) = \log p_\theta(\mathbf{y}' \mid \mathbf{z}^k) + \log p(\mathbf{z}^k) - \log q_\phi(\mathbf{z}^k \mid \mathbf{y}').$$

Minimizing the reflective likelihood term in Eq. 8 encourages the proposal q_ϕ to output a distinct approximate posterior distribution for each input $\mathbf{y}$. We conjecture this helps avoid the well-known issue of posterior collapse in the Variational Auto-Encoder (VAE) (Kingma & Welling, 2013)—this problem has been reported in various contexts (Burda et al., 2015; Bowman et al., 2015). We verify this later in our empirical study.

Choice of penalty level. The coefficient $\alpha(\cdot, \theta)$ in Eq. 4 and Eq. 8 induces a family of regularizers. Each choice of $\alpha(\cdot, \theta)$ leads to a different objective function and a different algorithm. In this paper we study two choices: (1) a global fixed coefficient $\alpha(\cdot, \theta) = \alpha_0$ and (2) a data-dependent coefficient:

$$\alpha(\cdot, \theta) = \begin{cases} \alpha_0 & \text{if } \log p_\theta(\mathbf{y}) \leq \log \hat{p}_\theta^{\text{refl}}(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

For supervised learning we replace $\log p_\theta(\mathbf{y})$ in Eq. 9 with $\log p_\theta(\mathbf{y} \mid \mathbf{x})$. For both (1) and (2) α_0 is a hyperparameter and gradients are computed by not differentiating through $\alpha(\cdot, \theta)$. In Section 4 we choose α_0 by evaluating performance on some held out validation set.

Connections to ranking losses. The choice of $\alpha(\cdot, \theta)$ in (2) corresponds to a ranking loss regularizer in the supervised learrning case. Ranking losses are ubiquitous in information retrieval. They are used as objective functions in many applications. For example Collobert & Weston (2008) use it for different natural language processing tasks. For that they define a scoring function $f(\cdot)$ and minimize

$$\mathcal{L}_{\text{ranking}} = \max(0, m - f(s^{\text{pos}}) + f(s^{\text{neg}})), \tag{10}$$

where m is some pre-specified margin, s^{pos} is a sample from the data—for example a sentence—,and s^{neg} is a negative sample—for example a sentence in the data where some words are replaced by other words.

Using the choice of $\alpha(\cdot, \theta)$ in Eq. (9), the proposed criterion $\mathcal{L}_{\text{RLL}}$ in Eq. (4) can be rewritten—up to a multiplicative constant—as

$$\mathcal{L}_{\text{RLL}} = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} \mathbb{E}_{\mathbf{y} \sim p_{\text{data}}(\mathbf{y} \mid \mathbf{x})} \left[(1 - \alpha_0) \log p_\theta(\mathbf{y} \mid \mathbf{x}) - \alpha_0 \max(0, -\log p_\theta(\mathbf{y} \mid \mathbf{x}) + \log p_\theta^{\text{refl}}(\mathbf{y})) \right].$$

To draw the connection to ranking consider one observed pair $(\mathbf{x}, \mathbf{y})$ and another single sample $\mathbf{x}'$ to approximate the reflective likelihood in Eq. (5) ($M = 1$). The objective is then

$$\mathcal{L}_{\text{RLL}} = (1 - \alpha_0) \log p_\theta(\mathbf{y} \mid \mathbf{x}) - \alpha_0 \max(0, -\log p_\theta(\mathbf{y} \mid \mathbf{x}) + \log p_\theta(\mathbf{y} \mid \mathbf{x}'))$$

The penalty here is a zero-margin ranking loss where the scoring function is

$$f(\mathbf{s}) = \log p_\theta(\mathbf{y} \mid \mathbf{s})$$

The ranking loss encourages the model to score $p_\theta(\mathbf{y} \mid \mathbf{x})$ higher than $p_\theta(\mathbf{y} \mid \mathbf{x}')$. This is in agreement with our motivation to prefer solutions which promote a strong dependence between inputs and outputs. As discussed in Collobert & Weston (2008), this leads to better performance in classification under imbalance. We verify this in Section 4 where we create several imbalanced versions of the MNIST dataset and compare classification performance against maximum likelihood.

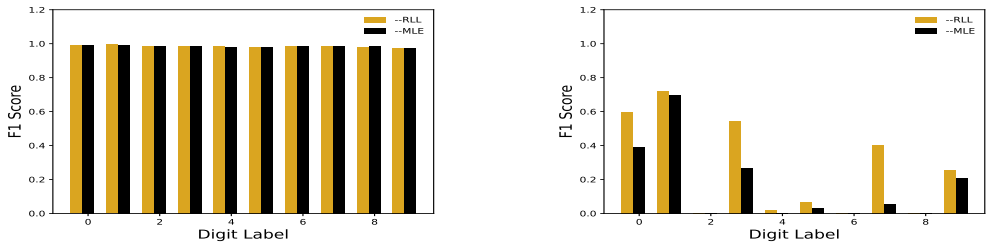
3 Related Work

Our work closely relates to two lines of work: penalized maximum likelihood and posterior inference in deep generative models.

Penalized maximum likelihood. Traditional maximum likelihood regularization methods such as the Lasso and L_2-norm regularization directly operate on the parameters of a model (Tibshirani, 1996; Hinton, 1987; Louizos et al., 2017). These regularizers penalize the magnitude of the parameters and correspond to specific prior distributions on the parameters from the Bayesian perspective. However, in the context of deep neural networks Pascanu et al. (2013) showed that such simple data-independent regularizers can cause difficulties in the learning procedure. In contrast several data-dependent regularizers have been proposed. As early as 1995, Bishop (1995) proposed to add noise to the input when training a neural network with stochastic gradient descent for maximum likelihood learning and showed this corresponds to a form of Tikhonov regularization. Several works have extended this to noise injection in the hidden units of a neural network (Srivastava et al., 2014; Maaten et al., 2013; Gal & Ghahramani, 2016; Wager et al., 2013; Dieng et al., 2018b). Our work relates to those data-dependent regularizers that are explicit—in the sense that the corresponding objective function can be written as the sum of the initial objective function and an additional regularization term.

Posterior inference in deep generative models. Latent variable models are ubiquitous in machine learning. They often involve intractable integrals that are approximated with Markov chain Monte Carlo or variational inference. Our work relates to variational methods for deep generative models (Kingma & Welling, 2013; Rezende et al., 2014). The problem that often occurs in these settings is the latent variables are driven independent to the observations thus rendering posterior inference meaningless. Several works have discussed this issue (Burda et al., 2015; Bowman et al., 2015; Hoffman & Johnson, 2016; Chen et al., 2016; Sønderby et al., 2016; Zhao et al., 2017; Alemi et al., 2018; Dieng et al., 2018a). In this paper we identify a possible cause to this problem and adopt a regularization approach to fix it.

Figure 1: Histogram of classification F1 scores for MLE and RLL. **Left:** Uniform distribution D1. **Right:** Imbalanced distribution D10. Performance of MLE and RLL on D1 is similar. However RLL outperforms MLE by a significant margin for the imbalanced distribution. This gain in performance comes from how well RLL performs on rare classes. For digits 2, 6, and 8 both MLE and RLL have 0 F1 scores.



4 APPLICATIONS

In this section we apply our proposed method to two different problems: image classification under imbalance and neural topic modeling. In classification under imbalance it is hard to learn features of rare classes. Because the objective in Eq. 4 promotes a stronger dependence between inputs and outputs we expect it to perform better than MLE in the presence of imbalance. We also consider an application in text modeling with deep generative latent variable models because the latent variable collapse problem is particularly severe in text modeling. Neural topic models suffer from this problem. One manifestation of that is all the dimensions of the topic matrix collapse to the same topic which in turn contains words that are not necessarily related to each other (Miao et al., 2016; Srivastava & Sutton, 2017).

On both of these applications we found our method yields better performance both quantitatively and qualitatively. It learns more useful features for rare classes in image classification on MNIST as evidenced by higher F1 scores (See Figure 1.) Finally it learns more meaningful latent variables as evidenced by lower perplexity on document completion (See Table 3.) We also found that the choice of $\alpha(\cdot, \theta)$ is dependent on the application. For classification we found a fixed schedule for alpha to perform best. For neural topic modeling we found an adaptive schedule to perform best. The reported results for RLL correspond to the best schedule for $\alpha(\cdot, \theta)$.

4.1 CLASSIFICATION UNDER IMBALANCE

In this section we study classification under imbalance. We use the MNIST dataset for this experiment. We hold out some of the training data as a validation set for hyperparameter search. We created several imbalanced versions of the training data using the class label distributions in Table 5 in the appendix. The distribution D1 is the uniform distribution and corresponds to perfect balance—each class in this setting has 5000 observations. The other distributions correspond to different imbalance levels.

Our classifier is a multilayer-CNN with max pooling. In more detail the first layer is a 2D convolutional layer with kernel dimension 5 and stride 1. The output of this layer is wrapped with a max pooling layer and a ReLU activation. The third layer is another 2D convolutional layer with kernel dimension 5 and stride 1 the output of which is passed to a max pooling layer and a ReLU activation. The final two layers are a sequence of linear maps and ReLU activations. Finally the predictive distribution is computed as the softmax of the output of the network.

The results are reported in Table 1 and Table 2. Under D1, maximum likelihood (MLE) and our method (RLL) perform similarly. Under all the imbalance settings, RLL outperforms MLE by a significant margin in terms of accuracy and F1 score. To assess performance on rare classes we visualize the histogram of the F1 scores on the uniform distribution D1 and on the most imbalanced distribution D10. Figure 1 illustrates the results. As can be seen in these histograms, RLL is particularly useful for countering the poor performance of MLE on rare classes. This confirms our hypothesis that regularizing maximum likelihood learning by minimizing the reflective likelihood of

Table 1: F1 scores and accuracies (the higher the better) for MLE and RLL on MNIST. The reported numbers are computed using the test set (which was unchanged). The RLL criterion outperforms MLE in every single setting.

Method	Metric	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
MLE	Acc	98.5	89.0	77.7	68.5	65.4	55.9	45.5	31.9	28.0	21.1
RLL	Acc	**98.6**	**92.0**	**83.3**	**70.2**	**71.7**	**59.1**	**48.8**	**35.6**	**33.8**	**31.1**
MLE	F1	98.5	84.5	69.3	57.9	57.8	43.9	32.2	19.5	20.6	17.5
RLL	F1	**98.6**	**92.6**	**80.9**	**60.7**	**65.9**	**48.4**	**37.2**	**22.8**	**27.4**	**27.0**

Table 2: F1 scores and accuracies (the higher the better) for MLE and RLL on MNIST. The reported numbers are computed on a **new test** set that was derived by applying the distributions in Table 5 (see appendix) to the original test set. The RLL criterion outperforms MLE in all settings but one where the two perform similarly.

Method	Metric	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
MLE	Acc	**99.2**	90.9	80.8	72.4	66.1	55.7	45.1	32.6	29.2	19.5
RLL	Acc	99.0	**93.9**	**85.1**	**74.5**	**74.1**	**57.3**	**48.1**	**37.5**	**34.1**	**28.8**
MLE	F1	**99.2**	87.0	73.1	62.7	57.5	43.7	31.2	20.7	22.8	15.9
RLL	F1	99.0	**94.3**	**83.3**	**66.2**	**68.8**	**45.9**	**36.1**	**24.1**	**27.8**	**24.8**

the data leads to a stronger dependence between inputs and outputs. In promoting a stronger dependence between inputs and outputs RLL captures useful features for rare classes. This is evidenced by higher F1 scores on these rare classes.

4.2 Neural Topic Modeling

Recently many works have focused on extending Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to neural networks. These neural topic models use the VAE architecture (Kingma & Welling, 2013; Rezende et al., 2014) and represent the posterior over topic distributions as a function over the output of a recognition network that takes a bag-of-word representation of the document (Miao et al., 2016; Srivastava & Sutton, 2017; Card et al., 2017).

We use a simple VAE architecture where the recognition network is a two-layer feed-forward neural network (MLP) with hyperbolic tangent activations. The output of this neural network is composed with a softmax activation to model the topic proportions and passed through a decoder. We represent the decoder as a three-layer MLP that maps the topic proportions to the vocabulary. The output matrix of this decoder is a topic matrix whose dimensions we visualize in Table 4. This particular form of neural topic models has been shown to be prone to posterior collapse (Srivastava & Sutton, 2017). We use it here as our model in order to single out the maximum likelihood objective function—the evidence lower bound (ELBO) in this case—as the cause of this posterior collapse.

For this experiment we used the 20NewsGroup benchmark dataset for topic modeling. 20NewsGroup is a collection of newsgroup documents, consisting of $11,314$ training and $7,531$ test articles. The vocabulary size for this corpus is $2,000$. We follow the standard preprocessing steps which involve tokenization, removal of some non-UTF-8 characters and English stop words.

For optimization we used stochastic optimization with Adam with a fixed learning rate of 0.002. We ran the model for 200 epochs and used 15 topics.

The results are presented in Table 3 and Table 4. Table 3 shows perplexity on the full test set but also the perplexity for document completion. Document completion consists in holding out some words for each document in the test set to compute the topic proportions and then evaluating perplexity on the remaining words of each document using the topic proportions learned with the held out words. It is a good way to assess the quality of the latent variables in topic modeling. We held out the first half of each document to compute the topic proportions and evaluated perplexity on the second half. As can be seen in Table 3 regularizing maximum likelihood with reflective likelihoods leads to better

Table 3: Perplexity (lower is better) and KL metric (higher is better) for the LDA-VAE topic model trained with RLL or MLE on the 20NewsGroup dataset. Note the difference in VAE (K=1) and IWAE(K=1) comes from the fact that for VAE we use a closed-form KL and not an importance sampling estimate of the KL. Here K denotes the number of posterior samples.

Criterion	K	Full PPL	Completion PPL	$\mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{y}) \,\|\, p(\mathbf{z}))$
MLE (ELBO)	1	841	911	2.3
MLE (IWAE)	1	820	884	2.8
RLL($\alpha_0 = 0.01$)	1	**809**	**875**	**3.0**
MLE (ELBO)	100	818	886	3.1
MLE (IWAE)	100	780	838	3.5
RLL($\alpha_0 = 0.01$)	100	**763**	**822**	**4.0**

Table 4: Top ten words of five randomly selected topics for different models trained with RLL and MLE on the 20NewsGroup dataset. Overall RLL learns better topics for the LDA-VAE model. Here M1 denotes LDA and M2 denotes the LDA-VAE model.

Setting	Topics
M1 (ELBO)	israel israeli jews arab state peace land jewish write policy pay tax insurance money write care year article health rate car drive engine buy write speed article light dealer driver offer sale condition mouse best old excellent tape month trade game team year win play season player fan write baseball
M2 (ELBO)	write article thanks want try need help buy work really write article thanks thing really look buy drive gun problem thanks run work write problem program software drive buy computer write article buy drive thanks problem car try want help armenians armenian kill turkish say child war government live attack
M2 (IWAE)	game play team player year write article better win baseball god christian jesus faith bible truth church christ believe christianity write article car thanks buy problem game bike team player armenians armenian turks turkish government israel state genocide attack serve thanks buy write car article phone appreciate drive sale run
M2 (RLL)	god christian jesus life faith church believe bible christ christianity game team play player win score year season hit toronto gun weapon say kill police come carry crime health criminal card drive windows mode driver problem pc disk printer scsi government key clipper chip secure encryption escrow law enforcement security

generalization. Furthermore Table 4 shows it fixes the collapse issue of maximum likelihood and leads to topics as good as those learned from LDA.

5 Conclusion

Models parameterized by deep neural networks have achieved state-of-the-art performance in many domains. They are often learned with maximum likelihood. However this learning criterion is prone to the problem of *input forgetting*. We identified a potential cause of this problem and proposed a new objective function for learning with deep generative models—including in the presence of latent variables. We studied this criterion on two applications—one in supervised learning and one in unsupervised learning—and found it led to better quantitative and qualitative performance than maximum likelihood.

REFERENCES

Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken elbo. In *International Conference on Machine Learning*, pp. 159–168, 2018.

Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Dallas Card, Chenhao Tan, and Noah A Smith. A neural framework for generalized topic models. *arXiv preprint arXiv:1705.09296*, 2017.

Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

KyungHyun Cho, Tapani Raiko, and Alexander T Ihler. Enhanced gradient and adaptive learning rate for training restricted boltzmann machines. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 105–112, 2011.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 160–167. ACM, 2008.

Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.

Adji B Dieng, Yoon Kim, Alexander M Rush, and David M Blei. Avoiding latent variable collapse with generative skip models. *arXiv preprint arXiv:1807.04863*, 2018a.

Adji B Dieng, Rajesh Ranganath, Jaan Altosaar, and David M Blei. Noisin: Unbiased regularization for recurrent neural networks. *arXiv preprint arXiv:1805.01500*, 2018b.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

RA Fisher. On an absolute criterion for fitting frequency curves. *Statistical Science*, 12(1):39–41, 1997.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304, 2010.

Geoffrey E Hinton. Learning translation invariant recognition in a massively parallel networks. In *International Conference on Parallel Architectures and Languages Europe*, pp. 1–13. Springer, 1987.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Matthew D Hoffman and Matthew J Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, 2016.

Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Daphne Koller, Nir Friedman, and Francis Bach. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*, 2017.

Laurens Maaten, Minmin Chen, Stephen Tyree, and Kilian Weinberger. Learning with marginalized corrupted features. In *International Conference on Machine Learning*, pp. 410–418, 2013.

David JC MacKay and Mark N Gibbs. Density networks. *Statistics and neural networks: advances at the interface. Oxford University Press, Oxford*, pp. 129–144, 1999.

Yishu Miao, Lei Yu, and Phil Blunsom. Neural variational inference for text processing. In *International Conference on Machine Learning*, pp. 1727–1736, 2016.

Radford M Neal. Connectionist learning of belief networks. *Artificial intelligence*, 56(1):71–113, 1992.

Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pp. 1310–1318, 2013.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. How to train deep variational autoencoders and probabilistic ladder networks. *arXiv preprint arXiv:1602.02282*, 2016.

Akash Srivastava and Charles Sutton. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*, 2017.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Stefan Wager, Sida Wang, and Percy S Liang. Dropout training as adaptive regularization. In *Advances in neural information processing systems*, pp. 351–359, 2013.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.

6 APPENDIX

Table 5: Class distributions using the MNIST dataset. There are 10 class—one class for each of the 10 digits in MNIST. The distribution D1 is uniform and the other distributions correspond to different imbalance settings as given by the proportions in the table. Note these proportions might not sum to one exactly because of rounding.

Dist	0	1	2	3	4	5	6	7	8	9
D1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
D2	$1e^{-3}$	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11
D3	$1e^{-3}$	$1e^{-3}$	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
D4	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	0.14	0.14	0.14	0.14	0.14	0.14	0.14
D5	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	0.17	0.17	0.17	0.17	0.17	0.17
D6	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	0.20	0.20	0.20	0.20	0.20
D7	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	0.25	0.25	0.25	0.25
D8	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	0.33	0.33	0.33
D9	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	0.49	0.49
D10	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	0.99